AD-A093 051 MARK RESOURCES INC MARINA DEL REY CA F/G 17/9

RFSS. FORTRAN PROGRAM FOR GENERATING REAL-TIME CLUTTER SEQUENCE--ETC(U)

OCT 78 R L MITCHELL DAAK40-78-C-0031

UNCLASSIFIED MRI-149-34 NL

1 OF 1

AD A 093051

END

DATE FILMED

1-81

DTIC

10

LEVEL II

U.S. ARMY MISSILE COMMAND

RFSS

FORTRAN PROGRAM FOR GENERATING REAL-TIME CLUTTER SEQUENCES

TECH NOTE 105-050

18 October 1978

14 MRI-TN-105-050

12

PREPARED FOR: RF SYSTEMS BRANCH (DRSMI-TDR) (R&D)
SYSTEMS SIMULATION DIRECTORATE
TECHNOLOGY LABORATORY
US ARMY MICOM
REDSTONE ARSENAL, AL 35809

Contract DAAK-40-78-C-0031

A

PREPARED BY: Dr. R. L. Mitchell
MARK RESOURCES, INC.
4676 ADMIRALTY WAY
SUITE 303
MARINA DEL REY, CA 90291
213-822-4955

MARK RESOURCES INCORPORATED

FORTRAN PROGRAM FOR GENERATING REAL-TIME CLUTTER SEQUENCES

MRI Report 149-34

R. L. Mitchell

18 October 1978

A

This Fortran program generates continuous time sequences of ground clutter that would be received on an airborne radar platform for a time-varying engagement geometry. The method is based on that in Reference 1. This program replaces earlier versions[2,3].

Signals are generated for the three monopulse channels. It is assumed that these signals will be radiated into specific positions on the receive antenna pattern so that each signal will be received by the proper channel and rejected by the others.

This program was written for a single range gate of a pulse-Doppler radar, although it is easily extended to multiple range gates (but not in real time with a single AP120B). The organization of this program is sketched, ~~in Figure 1~~. At the RFSS it is assumed that the real-time state of the environment will be set up in COMMON/C1/ and periodic calls will be made to CLUTTR in the Datacraft/6. The subroutines that actually generate the clutter sequences (FILL, CURVE, PARAB, CONVRT, TIMESQ, GAUSS4, FOURT) will be implemented in the AP120B. Their Fortran equivalents are included in this package.

A test program (TEST) is also included to compare results on different computers. The output of this program is a power spectral estimate of the clutter sequences generated under stationary conditions.

The geometry is shown in Figure 2, with all angles being used consistently throughout the program.

4676 ADMIRALTY WAY / SUITE 303 / MARINA DEL REY, CALIFORNIA 90291 / (213) 822-4955

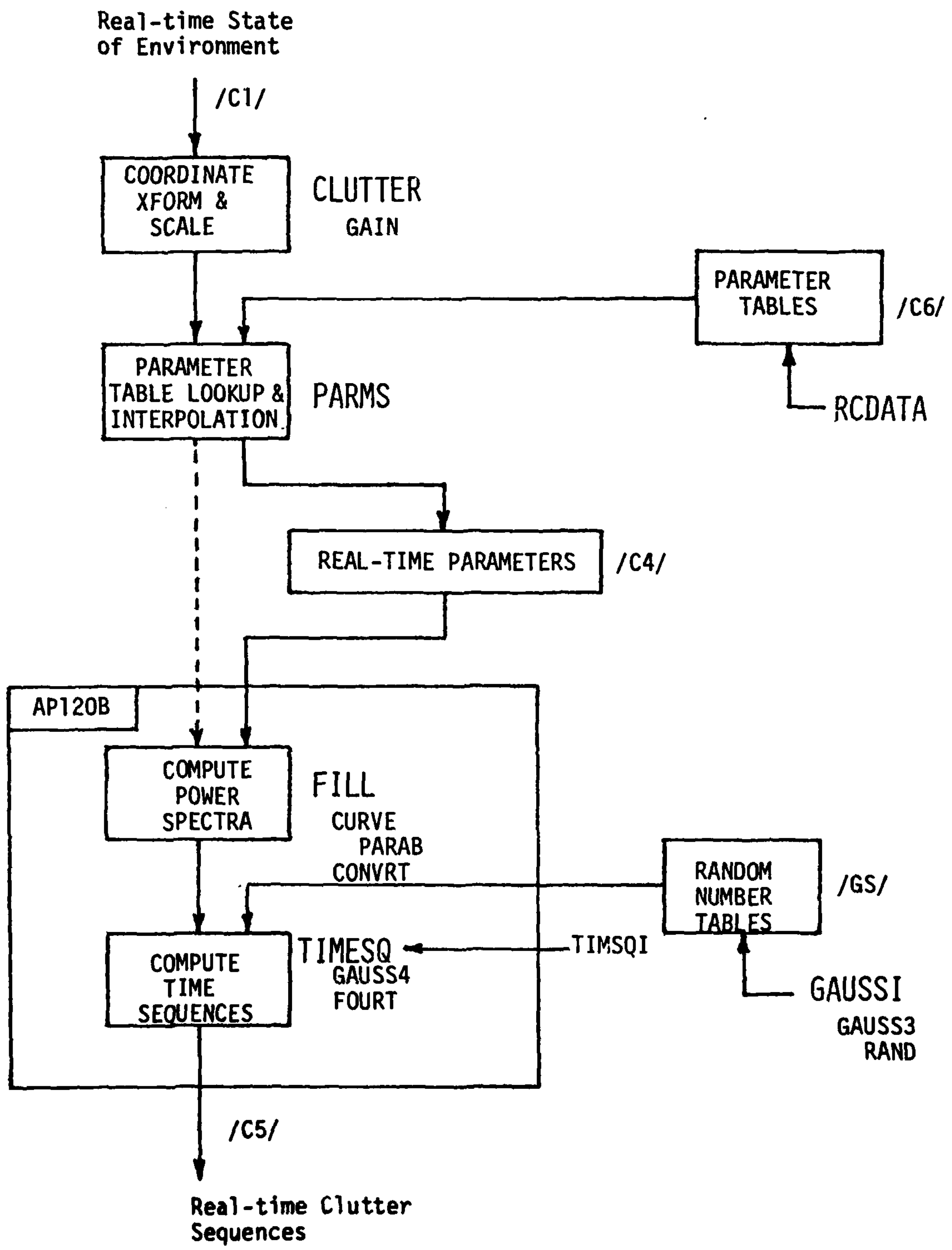

Figure 1. Block Diagram of Clutter Signal Program.

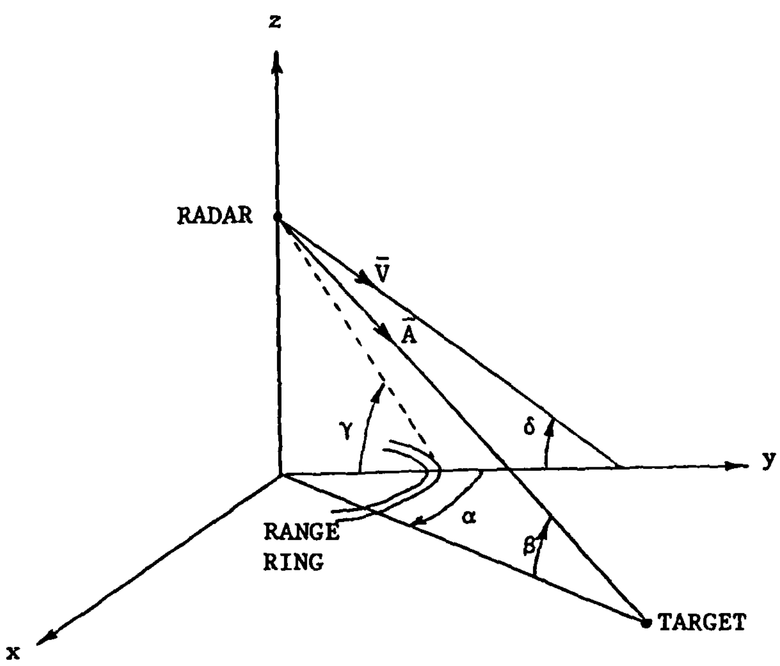

Figure 2. Engagement Geometry for Ground Clutter

PROGRAM SUMMARY

Main or Driver

TEST (for this example)

Initialization

RCDATA
TIMSQI
GAUSSI (GAUSS3,RAND)

Real-time Fortran

CLUTTR(GAIN)
PARMS(AINT)

Utility

PR
ACCUM
POWSPT
XMIT
FOURT

Real-time AP120B

FILL
CURVE
PARAB
CONVRT
TIMESQ
GAUSS4

```
      PROGRAM TEST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C TEST OF SUBROUTINE CLUTTR
C
C THE TEST CASE CORRESPONDS TO ALPHA=20.,BETA=0.,GAMMA=10.,DELTA=0.,  AND
C OMEGA=-22.5 (ALL DEGREES).
C
      DIMENSION SR(2240,3),SI(2240,3),S(200,3),WORK(600)
      COMMON /C1/ WL,H,V,R,FR,DE,OM,THO,PHO
      COMMON /C2/ PTGDSQ,SIGO,DR
      COMMON /C3/ NFFT,NOVLP,NF,A(64)
      COMMON /C5/ PEFF,
     1             XR1(320),XI1(320),XR2(320),XI2(320),XR3(320),XI3(320)
C
C THE FOLLOWING PARAMETERS MUST BE SET.....
C
      DATA WL,H,R,FR,V/.02,1736.,10000.,10000.,500./
      DATA DE,OM,THO,PHO/0.,-.3927,.3244,.1313/
      DATA PTGDSQ,DR,SIGO/1.,20.,.01/
      DATA NFFT,NOVLP/256,32/
C
C HERE WE INITIALIZE
C
      CALL RCDATA
      CALL TIMSQI
      CALL GAUSSI
C
C HERE WE GENERATE CONTIGUOUS TIME SEQUENCES IN THE SR AND SI ARRAYS
C
      L=1
      DO 20 I=1,10
      CALL CLUTTR
      CALL XMIT(NF,XR1,SR(L,1))
      CALL XMIT(NF,XR2,SR(L,2))
      CALL XMIT(NF,XR3,SR(L,3))
      CALL XMIT(NF,XI1,SI(L,1))
      CALL XMIT(NF,XI2,SI(L,2))
      CALL XMIT(NF,XI3,SI(L,3))
      L=L+NF
   20 CONTINUE
C
C HERE WE COMPUTE THE (AVERAGED) POWER SPECTRUM
C
      L=1
      DO 30 I=1,11
      CALL POWSPT(SR(L,1),SI(L,1),200,200,0.,WORK)
      CALL POWSPT(SR(L,2),SI(L,2),200,200,0.,WORK)
      CALL POWSPT(SR(L,3),SI(L,3),200,200,0.,WORK)
      CALL ACCUM(SR(L,1),S(1,1),200,I)
      CALL ACCUM(SR(L,2),S(1,2),200,I)
```

```
      CALL ACCUM(SR(L,3),S(1,3),200,I)
      L=L+200
   30 CONTINUE
      CALL PR(S,200,3)

      STOP
      END
```

```
      SUBROUTINE RCDATA
C
C THIS SUBROUTINE READS THE CLUTTER DATA BASE INTO COMMON /C6/.
C
      DIMENSION IW(4),W(22)
      COMMON /C6/ S(180),IS(4),A1(216),IA1(36),A2(180),IA2(36),E(216),
     1            IE(36)
      DATA IW/10HIS........,10HIA1.......,10HIA2.......,10HIE......../
      DATA W/10HS1........,10HS2........,10HS3........,10HS4........,
     1       10HS5........,10HA11.......,10HA12.......,10HA13.......,
     2       10HA14.......,10HA15.......,10HA1P.......,10HA22.......,
     3       10HA23.......,10HA24.......,10HA25.......,10HA2P.......,
     4       10HE1........,10HE2........,10HE3........,10HE4........,
     5       10HE5........,10HEP......../

      READ 102,S
      READ 101,IS
      READ 102,A1
      READ 101,IA1
      READ 102,A2
      READ 101,IA2
      READ 102,E
      READ 101,IE

      PRINT 200

      K1=1
      K2=36
      DO 10 L=1,5
      PRINT 202,W(L),(S(K),K=K1,K2)
      K1=K1+36
      K2=K2+36
   10 CONTINUE

      PRINT 201,IW(1),IS

      K1=1
      K2=36
      DO 20 L=6,11
      PRINT 202,W(L),(A1(K),K=K1,K2)
      K1=K1+36
      K2=K2+36
   20 CONTINUE

      PRINT 201,IW(2),IA1

      K1=1
      K2=36
      DO 30 L=12,16
      PRINT 202,W(L),(A2(K),K=K1,K2)
```

```
      K1=K1+36
      K2=K2+36
   30 CONTINUE

      PRINT 201,IW(3),IA2

      K1=1
      K2=36
      DO 40 L=17,22
      PRINT 202,W(L),(E(K),K=K1,K2)
      K1=K1+36
      K2=K2+36
   40 CONTINUE

      PRINT 201,IW(4),IE

      RETURN
  100 FORMAT(20F4.0)
  101 FORMAT(36I2)
  102 FORMAT(2(9F4.0,4X))
  200 FORMAT(1H1,50HGROUND CLUTTER DATA BASE READ BY SUBROUTINE RCDATA/)
  201 FORMAT(/1XA10,3X36I3)
  202 FORMAT(/1XA10,3X9F7.3/(14X9F7.3))
      END
```

```
      SUBROUTINE TIMSQI
C
C THIS SUBROUTINE PERFORMS INITIALIZATION FOR SUBROUTINE TIMESQ.
C
C THE INITIALIZATION PARAMETERS ARE.....
C
C      NFFT = SIZE OF FFT (=2**INTEGER)
C     NOVLP = NUMBER OF SAMPLES THAT THE FFTS OVERLAP (NFFT/8 IS OK)
C
C THE ARRAYS MUST BE DIMENSIONED AS LARGE AS.....
C
C           XR1,XI1,ETC          NFFT+NOVLP-1
C                     A          NOVLP-1
C
      COMMON /C3/ NFFT,NOVLP,NF,A(64)
      COMMON /C5/ PEFF,
     1           XR1(320),XI1(320),XR2(320),XI2(320),XR3(320),XI3(320)
      NF=NFFT-NOVLP
      N1=NOVLP-1
      N2=NFFT+N1
      DO 10 K=1,N1
      A(K)=K/FLOAT(NOVLP)
   10 CONTINUE
      CALL XMIT(-N2,0.,XR1)
      CALL XMIT(-N2,0.,XR2)
      CALL XMIT(-N2,0.,XR3)
      CALL XMIT(-N2,0.,XI1)
      CALL XMIT(-N2,0.,XI2)
      CALL XMIT(-N2,0.,XI3)
      RETURN
      END
```

```
      SUBROUTINE GAUSSI
C
C THIS SUBROUTINE FILLS ARRAYS TR AND TI WITH INDEPENDENT GAUSSIAN
C RANDOM NUMBERS FOR USE WITH SUBROUTINE GAUSS4.
C
C ARRAYS TR AND TI MUST BE DIMENSIONED AS LARGE AS NTR+NMAX AND NTI+NMAX
C RESPECTIVELY.
C
C IF N BELOW IS NOT EVEN, ADD 1 TO THE DIMENSION OF TI.
C
      COMMON /GS/ IRSET,NTR,NTI,NMAX,KR,KI,TR(1253),TI(1353)
      DATA NTR,NTI/997,1097/, NMAX/256/, KR,KI/1,1/
      N=NTR+NTI+2*NMAX
      DO 20 K=1,N,2
      CALL GAUSS3(TR(K),TR(K+1),1.)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GAUSS3(X,Y,P)
C
C THIS SUBROUTINE GENERATES RANDOM PAIRS FOR THESE DISTRIBUTIONS.....
C
C CALL GAUSS3(X,Y,P)....GAUSSIAN COMPONENTS OF AVERAGE POWER P
C CALL RANG3(X,Y,P).....RANDOM PHASOR COMPONENTS OF POWER P
C
      DATA Z/0./
C
      IF(P) 30,20,10
   10 E=SQRT(-P*ALOG(RAND(Z)))
      GO TO 15
      ENTRY RANG3
      IF(P) 30,20,12
   12 E=SQRT(P)
   15 A=RAND(Z)
      A=A+A-1.
      B=RAND(Z)
      B=B+B-1.
      A2=A*A
      B2=B*B
      C=A2+B2
      IF(C.GT.1.) GO TO 15
      X=E*(A2-B2)/C
      Y=E*2.*A*B/C
      RETURN
   20 X=0.
      Y=0.
      RETURN
   30 STOP 44
      END
```

```
      FUNCTION RAND(Z)
C FOR TEST PURPOSES ONLY.
      DIMENSION A(8)
      DATA A/.1,.2,.3,.4,.5,.6,.7,.8/
      DATA K/1/
      RAND=A(K)
      K=K+1
      IF(K.GT.8) K=1
      RETURN
      END
```

```
      SUBROUTINE CLUTTR
C
C IN THIS SUBROUTINE WE GENERATE THE CLUTTER MODULATION SIGNALS FOR ONE
C RANGE GATE ON EACH OF THREE MONOPULSE CHANNELS.
C
C THE BACKSCATTER COEFFICIENT OF THE GROUND IS ASSUMED TO VARY AS THE
C SINE OF THE GRAZING ANGLE.
C
C THE REAL-TIME PARAMETERS ON INPUT ARE.....
C
C        WL = WAVELENGTH
C         H = PLATFORM ALTITUDE
C         V = PLATFORM VELOCITY
C         R = RANGE TO RANGE GATE OF INTEREST (UNAMBIGUOUS)
C        FR = EQUIVALENT SAMPLE RATE IN RECEIVER (PROCESSING BAND)
C        DE = DELTA, THE MISSILE DIVE ANGLE
C        OM = OMEGA, THE MISSILE ROLL ANGLE (CCW IS POSITIVE)
C       THO = THETA-ZERO, THE AZIMUTH ANGLE OF THE ANTENNA BORESIGHT
C       PHO = PHI-ZERO, THE ELEVATION ANGLE OF THE ANTENNA BORESIGHT
C
C THE NONREAL-TIME PARAMETERS ON INPUT ARE.....
C
C    PTGDSQ = PRODUCT OF TRANSMIT POWER, GAIN, SQUARE OF CHAMBER LENGTH
C        DR = RANGE RESOLUTION
C      SIGO = GROUND BACKSCATTER COEFFICIENT AT 30-DEG GRAZING ANGLE
C
C THE REAL-TIME OUTPUT FOR THAT PORTION COMPUTED IN THE DATACRAFT IS
C THROUGH /C4/, AND IT IS COMPUTED IN SUBROUTINE PARMS.
C
C SOME OF THE OTHER QUANTITIES USED IN THIS SUBROUTINE ARE.....
C
C        AL = ALPHA, THE AZIMUTH ANGLE OF THE BEAM AXIS IN GROUND COORD.
C        BE = BETA, THE ELEVATION ANGLE OF THE BEAM AXIS IN GROUND COORD.
C        GA = GAMMA, THE GRAZING ANGLE AT THE RANGE RING ON THE GROUND
C
C ALL ANGLES ARE IN RADIANS AND ALL DISTANCES IN METERS.
C
      COMMON /C1/ WL,H,V,R,FR,DE,OM,THO,PHO
      COMMON /C2/ PTGDSQ,SIGO,DR
      COMMON /C3/ NFFT
      DATA FCTR/1./
C
C COMPUTE GEOMETRY AND DOPPLER PARAMETERS
C
      SGA=H/R
      CGA=SQRT(1.-SGA**2)
      GA=ATAN(SGA/CGA)
      CPHO=COS(PHO)
      SPHO=SIN(PHO)
      CTHO=COS(THO)
```

```
      STHO=SIN(THO)
      COM=COS(OM)
      SOM=SIN(OM)
      CDE=COS(DE)
      SDE=SIN(DE)
      AX=CPHO*STHO
      AY=CPHO*CTHO
      AZ=SPHO
      BX=AX*COM-AZ*SOM
      BY=AY
      BZ=AX*SOM+AZ*COM
      CX= BX
      CY= BY*CDE+BZ*SDE
      CZ=-BY*SDE+BZ*CDE
      AL=ATAN(CX/CY)
      BE=ATAN(-CZ/SQRT(1.-CZ**2))
      PEFF=.013*PTGDSQ*(2.*SIGO*SGA)*DR*GAIN(GA-BE)/R**3
      FMAX=2.*V/WL
      FMC =FMAX*(CGA*CDE+SGA*SDE)
      FSCL=.9848*FMAX/FMC
      IAMB=FMC/FR
      FNP1=(IAMB-1)*FR*FSCL/FMAX
      DFNP=FNP1/FLOAT((IAMB-1)*NFFT)
      FNP1=.9848-FCTR*(.9848-FNP1)
      DFNP=FCTR*DFNP
      ALD=57.3*AL
      OMD=57.3*OM
      IF(ALD.GT.0.) GO TO 10
      OMD=360.-OMD
      ALD=-ALD
   10 OMDP=OMD
      IF(OMD.GT.315.) OMDP=OMDP-360.
      ICASE=(OMDP+45.)/90.
      OMDP=OMDP-ICASE*90.
      IROT=0
      IF(ICASE.EQ.1.OR.ICASE.EQ.3) IROT=1
C                 )/4.21E,4.9F3,1.9F2,1.8F3,4.8F3,4I3/.....RTTULC H21/(TAMROF 001
C                                                        FFEP,PNFD,1PNF          1
C             ,LCSF,CMF,XAMF,PDMO,DMO,DLA,EB,LA,AG,TORI,ESACI,BMAI,001 TNIRP
C
C COMPUTE REAL-TIME PARAMETERS
C
      CALL PARMS(ALD,OMDP,FNP1,DFNP)
C
C GENERATE REAL-TIME SPECTRA
C
      IROT=0
      IF(ICASE.EQ.1.OR.ICASE.EQ.3) IROT=1
      CALL FILL(IROT)
      RETURN
      END
```

```
      FUNCTION GAIN(PHI)
C
C HERE WE COMPUTE THE 2-WAY POWER GAIN BY TABLE LOOKUP.   PHI IS THE
C ANGLE IN RADIANS.
C
C        NB = NUMBER OF SAMPLES IN TABLE
C        DB = SAMPLE SPACING (DEG)
C
C THE FIRST SAMPLE IS AT PHI=0.
C
      DIMENSION BEAM(101)
      DATA NB/101/,DB/.5/
      DATA A/.00389/,B/2.30/
      DATA NO/0/
      IF(NO.GT.0) GO TO 30
      NO=1
      DO 20 K=1,NB
      BEAM(K)=EXP(-4.*A*((K-1)*DB)**B)
   20 CONTINUE
   30 E=57.3*ABS(PHI)
      IE=E/DB+1.5
      GAIN=0.
      IF(IE.LE.NB) GAIN=BEAM(IE)
      RETURN
      END
```

```
      SUBROUTINE PARMS(ALPHA,OMEGA,F1,DF)
C
C IN THIS SUBROUTINE WE COMPUTE THE REAL-TIME PARAMETERS THAT WILL BE
C SENT TO THE AP120B.  THE PARAMETERS ARE DETERMINED BY INTERPOLATION
C WITHIN THE DATA BASE STORED IN COMMON /C6/.
C
C THE INPUT IS.....
C
C        ALPHA = ABSOLUTE VALUE OF AZIMUTH ANGLE (DEG)
C        OMEGA = MISSILE ROLL ANGLE (DEG,   -45..LE.OMEGA.LE.45.)
C           F1 = NORMALIZED DOPPLER OF FIRST SAMPLE
C           DF = SAMPLE SPACING IN DOPPLER
C
C THE OUTPUT IS THROUGH COMMON /C4/.....
C
C        IS,SPK......SUM SPECTRUM PARAMETERS
C        IA1,A1PK....DEL-AZ SPECTRUM PARAMETERS, LEFT HALF
C        IA2,A2PK....DEL-AZ SPECTRUM PARAMETERS, RIGHT HALF
C        IE,EPK......DEL-EL SPECTRUM PARAMETERS
C        PSL.........SIDELOBE LEVEL (SEE SUBROUTINE FILL)
C
C SKP,A1PK,A2PK,EPK,AND PSL MUST BE IN THE SAME UNITS (NEPERS).
C
C                                                              TNIA LANRETXE
      COMMON /C4/ IS(6),IA1(6),IA2(6),IE(6),SPK,A1PK,A2PK,EPK,PSL
      COMMON /C6/ S1(36),S2(36),S3(36),S4(36),S5(36),IST(4),
     1       A11(36),A12(36),A13(36),A14(36),A15(36),A1P(36),IA1T(36),
     2               A22(36),A23(36),A24(36),A25(36),A2P(36),IA2T(36),
     3            E1(36),E2(36),E3(36),E4(36),E5(36),EP(36),IET(36)
      COMMON /CINT/ K1,K2,K3,K4,HOM,HAL
      DATA NAL,NOM/4,9/
      DATA DAL,DOM/10.,11.25/
      ISAMP(A)=(AINT(A)-F1)/DF+1.5
C
C HERE WE FIND THE PARAMETERS FOR INTERPOLATION
C
      AL=ALPHA/DAL
      IAL=AL
      HAL=AL-IAL
      IAL=IAL+1
      IF(IAL.GE.NAL) HAL=1.
      IF(IAL.GE.NAL) IAL=NAL-1
      OM=(OMEGA+45.)/DOM
      IOM=OM
      HOM=OM-IOM
      IOM=IOM+1
      IF(IOM.GE.NOM) HOM=1.
      IF(IOM.GE.NOM) IOM=NOM-1
      K1=(IAL-1)*NOM+IOM
      K2=K1+1
```

```
      K3=K1+NOM
      K4=K2+NOM
      KAL=ALPHA/DAL+1.5
      KAL=MIN0(KAL,NAL)
      KOM=(OMEGA+45.)/DOM+1.5
      K=(KAL-1)*NOM+KOM
C
C HERE WE COMPUTE THE REAL-TIME PARAMETERS
C
      IS(1) =ISAMP(S1)
      IS(2) =ISAMP(S2)
      IS(3) =ISAMP(S3)
      IS(4) =ISAMP(S4)
      IS(5) =ISAMP(S5)
      IA1(1)=ISAMP(A11)
      IA1(2)=ISAMP(A12)
      IA1(3)=ISAMP(A13)
      IA1(4)=ISAMP(A14)
      IA1(5)=ISAMP(A15)
      IA2(1)=IA1(5)
      IA2(2)=ISAMP(A22)
      IA2(3)=ISAMP(A23)
      IA2(4)=ISAMP(A24)
      IA2(5)=ISAMP(A25)
      IE(1) =ISAMP(E1)
      IE(2) =ISAMP(E2)
      IE(3) =ISAMP(E3)
      IE(4) =ISAMP(E4)
      IE(5) =ISAMP(E5)
      SPK =40.
      PSL =0.
      A1PK=AINT(A1P)
      A2PK=AINT(A2P)
      EPK =AINT(EP)
      IS(6) =IST(KAL)
      IA1(6)=IA1T(K)
      IA2(6)=IA2T(K)
      IE(6) =IET(K)
C
C CONVERT DB TO NEPERS (AMPLITUDE)
C
      SPK =.11513*SPK
      A1PK=.11513*A1PK
      A2PK=.11513*A2PK
      EPK =.11513*EPK
      PSL =.11513*PSL

      RETURN
      END
```

```
      FUNCTION AINT(A)
C
C INTERPOLATION FOR SUBROUTINE PARMS
C
      DIMENSION A(1)
      COMMON /CINT/ K1,K2,K3,K4,HOM,HAL
      A1=(1.-HOM)*A(K1)+HOM*A(K2)
      A2=(1.-HOM)*A(K3)+HOM*A(K4)
      AINT=(1.-HAL)*A1+HAL*A2
      RETURN
      END
```

```
      SUBROUTINE PR(A,NX,NY)
C
C IN THIS SUBROUTINE WE PRINT THE 2-DIMENSIONAL ARRAY A WITH THE FORMAT
C EQUIVALENT TO THE DIMENSION A(NX,NY).
C
C THE PRINTING IS ON FILE LU, WHICH MUST BE EQUIPPED.
C
      DIMENSION A(1)
      COMMON /MESGE/ MFLAG
      DATA LU/6/
      IF(MFLAG.NE.1) WRITE (LU,100)
      MFLAG=0
      DO 20 J=1,NY
      L1=(J-1)*NX+1
      L2=L1+9
      L2MAX=J*NX
      L2=MINO(L2,L2MAX)
      DO 20 I=1,NX,10
      IF(I.LE.1) WRITE (LU,101) J,(A(K),K=L1,L2)
      IF(I.GT.1) WRITE (LU,102)   (A(K),K=L1,L2)
      L1=L1+10
      L2=MINO(L2+10,L2MAX)
   20 CONTINUE
      RETURN
  100 FORMAT(1H1)
  101 FORMAT(/I4,2X10E12.4)
  102 FORMAT(6X10E12.4)
      END
```

```
      SUBROUTINE ACCUM(X,Y,N,IREP)
C
C IN THIS SUBROUTINE WE ACCUMULATE AN ARRAY.  AT ALL TIMES ARRAY Y
C REPRESENTS A RUNNING AVERAGE.
C
      DIMENSION X(1),Y(1)
      IF(IREP.EQ.1) GO TO 30
      REP=IREP
      DO 20 K=1,N
      Y(K)=((REP-1.)*Y(K)+X(K))/REP
   20 CONTINUE
      RETURN
   30 CALL XMIT(N,X,Y)
      RETURN
      END
```

```
      SUBROUTINE POWSPT(XR,XI,NIN,NOUT,ALPHA,W)
C
C IN THIS SUBROUTINE WE COMPUTE THE POWER SPECTRUM OF THE COMPLEX TIME
C SEQUENCE IN THE ARRAY-PAIR (XR,XI) OF LENGTH NIN.  THE POWER SPECTRUM
C IS RETURNED IN ARRAY XR, AND IT IS NOW OF LENGTH NOUT.
C
C THE SAMPLE SPACING OF THE POWER SPECTRUM IS 1/NOUT OF THE REPETITION
C FREQUENCY.
C
C A COSINE-ON-A-PEDESTAL WEIGHTING IS APPLIED TO THE INPUT SAMPLES.
C ALPHA IS THE RATIO OF THE WEIGHTING FUNCTION AT THE EDGE TO THE
C CENTER.  ALPHA=.08 FOR HAMMING AND ALPHA=1.0 FOR UNIFORM WEIGHTING.
C
C ARRAY W IS A WORKING ARRAY AND IT MUST BE DIMENSIONED AS LARGE AS.....
C
C              NIN            IF NOUT.EQ.2**INTEGER
C              NIN+2*NOUT     IF NOUT.NE.2**INTEGER
C
C THE WEIGHTS ARE NORMALIZED SO THAT THE SUM IS UNITY.
C
      DIMENSION XR(1),XI(1),W(1)
      DATA TWOPI/6.2831853/
      DATA NO/0/
      IF(NO.GT.0) GO TO 25
      NO=1
      N=NOUT-NIN
      A=(1.+ALPHA)/2.
      B=(1.-ALPHA)/2.
      CN=(NIN+1)/2.
      XN=NIN
      DO 10 K=1,NIN
      W(K)=A+B*COS(TWOPI*(K-CN)/XN)
   10 CONTINUE
      SUM=0.
      DO 15 K=1,NIN
      SUM=SUM+W(K)
   15 CONTINUE
      WNORM=SUM
      DO 20 K=1,NIN
      W(K)=W(K)/WNORM
   20 CONTINUE
   25 IF(N) 60,35,30
   30 CALL XMIT(-N,0.,XR(NIN+1))
      CALL XMIT(-N,0.,XI(NIN+1))
   35 DO 40 K=1,NIN
      XR(K)=XR(K)*W(K)
      XI(K)=XI(K)*W(K)
   40 CONTINUE
      CALL FOURT(XR,XI,NOUT,1,1,1,W(NIN+1),W(NIN+NOUT+1))
      DO 50 K=1,NOUT
```

```
      XR(K)=XR(K)**2+XI(K)**2
   50 CONTINUE
      RETURN
   60 PRINT 100,NIN,NOUT
      STOP
  100 FORMAT(/1X49H***ERROR IN POWSPT , NOUT IS SMALLER THAN NIN ***/
     1     /10X4HNIN=I6,10X5HNOUT=I6)
      END
```

```
      SUBROUTINE XMIT(N,A,B)
C
C IN THIS SUBROUTINE WE EITHER TRANSMIT ARRAY A TO ARRAY B (IF N.GT.0)
C OR WE TRANSMIT THE CONSTANT A TO ARRAY B (IF N.LT.0).  IN EITHER CASE
C THE ARRAY LENGTH IS IABS(N).
C
C THIS SUBROUTINE SHOULD BE WRITTEN IN ASSEMBLY LANGUAGE
C
      DIMENSION A(1),B(1)
      IF(N) 10,20,25
   10 NN=-N
      AA=A(1)
      DO 15 K=1,NN
      B(K)=AA
   15 CONTINUE
   20 RETURN
   25 DO 30 K=1,N
      B(K)=A(K)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FILL(IROT)
C
C IN THIS SUBROUTINE WE GENERATE THE REAL-TIME CLUTTER MODULATION
C SIGNALS ON THE THREE MONOPULSE CHANNELS (FOR ONE RANGE GATE ONLY).
C THIS SUBROUTINE, AS WELL AS SUBROUTINES CURVE, CONVRT, PARAB, TIMESQ,
C GAUSS4, AND FFT2, WILL BE IMPLEMENTED ON THE AP120B.
C
C THE REAL-TIME INPUT IS THROUGH /C4/.....
C
C       IS,SPK......SUM SPECTRUM PARAMETERS
C       IA1,A1PK....DEL-AZ SPECTRUM PARAMETERS, LEFT HALF
C       IA2,A2PK....DEL-AZ SPECTRUM PARAMETERS, RIGHT HALF
C       IE,EPK......DEL-EL SPECTRUM PARAMETERS
C       PSL.........SIDELOBE LEVEL (SEE BELOW)
C
C THE REAL-TIME OUTPUT IS THROUGH /C5/.....
C
C      PEFF = EFFECTIVE POWER RADIATED FROM RFSS ARRAY (FROM SUB. CLUTTR)
C       XR1 = REAL ARRAY CONTAINING SUM CHANNEL MODULATION SIGNAL
C       XI1 = IMAG ARRAY CONTAINING SUM CHANNEL MODULATION SIGNAL
C       XR2 = REAL ARRAY CONTAINING DAZ CHANNEL MODULATION SIGNAL
C       XI2 = IMAG ARRAY CONTAINING DAZ CHANNEL MODULATION SIGNAL
C       XR3 = REAL ARRAY CONTAINING DEL CHANNEL MODULATION SIGNAL
C       XI3 = IMAG ARRAY CONTAINING DEL CHANNEL MODULATION SIGNAL
C
C IROT IS A ROTATION FLAG.  IF IROT.GT.0 THE AZ AND EL ARRAYS ARE
C INTERCHANGED.
C
C THIS PROGRAM GENERATES A CONSTANT-LEVEL SIDELOBE SHOULDER OF (SPK-PSL)
C DB BELOW THE SUM SPECTRUM PEAK.
C
C ARRAYS S,A, AND E ARE FOR TEMPORARY STORAGE OF THE SUM, AZ- AND EL-
C DIFFERENCE SPECTRA.  THEY MUST BE DIMENSIONED AS LARGE AS 2*NFFT.
C
C THE DIMENSIONED ARRAYS IN COMMON /C5/ MUST ACCOMMODATE NFFT+NOVLP
C WORDS (SEE TIMESQ).
C
      COMMON /C3/ NFFT
      COMMON /C4/ IS(6),IA1(6),IA2(6),IE(6),SPK,A1PK,A2PK,EPK,PSL
      COMMON /C5/ PEFF,
     1            XR1(320),XI1(320),XR2(320),XI2(320),XR3(320),XI3(320)
      COMMON /C7/ S(512),A(512),E(512)
      COMMON /LIM/ NFFT2
      NFFT2=NFFT*2
C
C INITIALIZE ARRAYS TO SIDELOBE LEVEL
C
      CALL XMIT(-NFFT2,PSL,S)
      CALL XMIT(-NFFT2,PSL,A)
      CALL XMIT(-NFFT2,PSL,E)
```

```
C
C GENERATE LOG-AMPLITUDE SPECTRA
C
      CALL CURVE(S,IS,SPK)
      CALL CURVE(A,IA1,A1PK)
      CALL CURVE(A,IA2,A2PK)
      CALL CURVE(E,IE,EPK)
C
C CONVERT TO LINEAR AMPLITUDE AND FOLDOVER
C
      SNORM=0.
      CALL CONVRT(S,SNORM)
      CALL CONVRT(A,SNORM)
      CALL CONVRT(E,SNORM)
C
C GENERATE TIME SEQUENCES
C
      CALL TIMESQ(S,XR1,XI1)
      IF(IROT.GT.0) GO TO 35
      CALL TIMESQ(A,XR2,XI2)
      CALL TIMESQ(E,XR3,XI3)
      GO TO 40
   35 CALL TIMESQ(A,XR3,XI3)
      CALL TIMESQ(E,XR2,XI2)
   40 RETURN
      END
```

```
      SUBROUTINE CURVE(S,II,PK)
C
C THIS SUBROUTINE GENERATES A CURVE IN ARRAY S BETWEEN SAMPLES II(1) AND
C II(5).  THE CURVE IS DEFINED IN FOUR SEGMENTS BETWEEN FIVE POINTS
C WHERE THE POINTS ARE GIVEN BY.....
C
C                     SAMPLE          VALUE
C
C                     II(1)           0.
C                     II(2)           PK/2.
C                     II(3)           PK
C                     II(4)           PK/2.
C                     II(5)           0.
C
C II(6)=ITYP IS A PARAMETER THAT DEFINES THE TYPE OF SEGMENTS.....
C
C                     ITYP            SEGMENTS
C
C                     0               LINE,LINE,LINE,LINE
C                     1               LINE,PARAB,LINE,LINE
C                     2               LINE,PARAB,PARAB,LINE
C
C THE PARABOLA IS ALWAYS CONSTRAINED TO HAVE ZERO SLOPE AT THE CENTER
C POINT.
C
      DIMENSION S(1),II(6)
      COMMON /ABC/ A,B,C

      IF(PK.LE.0.) RETURN
      P2=.5*PK
      ITYP=II(6)
C
C COMPUTE FIRST SEGMENT
C
      C=0.
      B=P2/AMAX0(II(2)-II(1),1)
      A=-B*II(1)
      CALL PARAB(S,II(1),II(2))
C
C COMPUTE SECOND SEGMENT
C
      IF(ITYP.GT.0) GO TO 20
      C=0.
      B=P2/AMAX0(II(3)-II(2),1)
      A=P2-B*II(2)
      GO TO 25
   20 C=-P2/AMAX0(II(3)-II(2),1)**2
      B=-2.*C*II(3)
      A=P2-B*II(2)-C*II(2)**2
   25 CALL PARAB(S,II(2)+1,II(3))
```

```
C
C COMPUTE THIRD SEGMENT
C
      IF(ITYP.GT.1) GO TO 30
      C=0.
      B=-P2/AMAXO(II(4)-II(3),1)
      A=PK-B*II(3)
      GO TO 35
   30 C=-P2/AMAXO(II(4)-II(3),1)**2
      B=-2.*C*II(3)
      A=PK-B*II(3)-C*II(3)**2
   35 CALL PARAB(S,II(3)+1,II(4))
C
C COMPUTE FOURTH SEGMENT
C
      C=0.
      B=-P2/AMAXO(II(5)-II(4),1)
      A=P2-B*II(4)
      CALL PARAB(S,II(4)+1,II(5))

      RETURN
      END
```

```
      SUBROUTINE PARAB(S,II1,II2)
C
C THIS SUBROUTINE GENERATES A PARABOLA OF THE FORM.....
C
C               S(I) = A + B*I + C*I**2
C
      DIMENSION S(1)
      COMMON /ABC/ A,B,C
      COMMON /LIM/ LIM
      I1=MAX0(II1,1)
      I2=MIN0(II2,LIM)
      IF(I1.GT.I2) RETURN
      DO 20 I=I1,I2
      S(I)=(C*I+B)*I+A
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONVRT(S,SNORM)
C
C IN THIS SUBROUTINE WE CONVERT THE SPECTRUM IN ARRAY S OF LOG-AMPLITUDE
C UNITS TO LINEAR-AMPLITUDE UNITS.  IN ADDITION, TWO PRF INTERVALS ARE
C FOLDED (SUMMED).  THE INPUT S-ARRAY IS OF LENGTH 2*NFFT AND THE OUTPUT
C S-ARRAY IS OF LENGTH NFFT.  AN APPROXIMATION IS USED IN THE DO-20 LOOP
C FOR A FASTER COMPUTATION.
C
C FOR THE SUM SPECTRUM SET SNORM=ZERO ON INPUT (DO NOT USE THE CONSTANT
C 0., HOWEVER).  THE NORMALIZATION FACTOR SNORM WILL BE RETURNED.  FOR
C THE DIFFERENCE SPECTRA USE THIS SAME VALUE ON INPUT.
C
      DIMENSION S(1)
      COMMON /C3/ NFFT
      DO 20 K=1,NFFT
      S(K)=EXP(S(K)+S(K+NFFT))
   20 CONTINUE
      IF(SNORM.GT.0.) GO TO 35
      SUM=0.
      DO 30 K=1,NFFT
      SUM=SUM+S(K)**2
   30 CONTINUE
      SNORM=1./SQRT(SUM)
   35 DO 40 K=1,NFFT
      S(K)=S(K)*SNORM
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TIMESQ(SA,XR,XI)
C
C IN THIS SUBROUTINE WE GENERATE A CORRELATED TIME SEQUENCE THAT HAS AN
C AMPLITUDE SPECTRAL FUNCTION DESCRIBED BY ARRAY SA OF LENGTH NFFT.  THE
C METHOD USED IS THE ON-LINE FFT APPROACH DESCRIBED IN SECTION 6 OF MRI
C REPORT 132-44, WHERE SUCCESSIVE CALLS TO TIMESQ WILL CREATE CONTIGUOUS
C TIME SEQUENCES BASED ON OVERLAPPING FFTS IN THE FREQUENCY DOMAIN.
C
C ON INPUT.....
C
C      NFFT = SIZE OF FFT (=2**INTEGER)
C     NOVLP = NUMBER OF SAMPLES THAT THE FFTS OVERLAP (NFFT/8 IS OK)
C     SA(K) = SPECTRAL AMPLITUDE OF KTH SAMPLE, K=1,...,NFFT
C
C ON OUTPUT.....
C
C     XR(K) = KTH SAMPLE OF IN-PHASE SIGNAL, K=1,...,NF
C     XI(K) = KTH SAMPLE OF QUADRATURE SIGNAL, K=1,...,NF
C
C WHERE  NF=NFFT-NOVLP.
C
C THE ARRAYS MUST BE DIMENSIONED AS LARGE AS.....
C
C                         SA          NFFT
C                         XR          NFFT+NOVLP-1
C                         XI          NFFT+NOVLP-1
C                         A           NOVLP-1
C
C NOTE THAT THE INFORMATION IS STORED IN THE XR AND XI ARRAYS FROM
C NFFT+1 TO NFFT+NOVLP+1 THAT WILL BE USED ON THE NEXT CALL.  PRIOR TO
C THE FIRST CALL TO TIMESQ THE INITIALIZATION SUBROUTINE (TIMSQI) MUST
C BE CALLED.
C
C THE WORK ARRAYS WRKR AND WRKI NEED BE DIMENSIONED ONLY IF NFFT.NE.
C 2**INTEGER, IN WHICH CASE THE DIMENSION SIZE MUST BE NFFT.
C
      DIMENSION WRKR(160),WRKI(160)
      DIMENSION SA(1),XR(1),XI(1)
      COMMON /C3/ NFFT,NOVLP,NF,A(1)
      CALL GAUSS4(SA,XR,XI,NFFT)
      CALL FOURT(XR,XI,NFFT,1,0,1,WRKR,WRKI)
      N1=NOVLP-1
      DO 20 K=1,N1
      XR(K)=A(K)*(XR(K)-XR(K+NFFT))+XR(K+NFFT)
      XI(K)=A(K)*(XI(K)-XI(K+NFFT))+XI(K+NFFT)
      XR(K+NFFT)=XR(K+NFFT-NOVLP)
      XI(K+NFFT)=XI(K+NFFT-NOVLP)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GAUSS4(A,XR,XI,N)
C
C GENERATES RANDOM PHASOR COMPONENTS (XR(K),XI(K)) OF ZERO MEAN WITH THE
C AVERAGE POWER OF  XR(K)**2+XI(K)**2  GIVEN BY A(K)**2 FOR K=1,...,N.
C
C THE RANDOM NUMBERS ARE ACCESSED FROM RECIRCULATING TABLES.  NTR AND
C NTI ARE THE CIRCULATION PARAMETERS AND NMAX=MAX(N).  WE SHOULD CHOOSE
C NTR AND NTI RELATIVELY PRIME.
C
C ARRAYS TR AND TI MUST BE DIMENSIONED AS LARGE AS NTR+NMAX AND NTI+NMAX
C RESPECTIVELY.
C
C IF IRSET=1 IN COMMON /GS/, THE POINTERS WILL BE RESTORED ON EXIT TO
C THEIR ORIGINAL VALUES WHEN GAUSS4 WAS ENTERED.  THIS MEANS THAT THE
C SAME GAUSSIAN SEQUENCE (EXCEPT FOR THE WEIGHTS A) WILL BE GENERATED ON
C THE NEXT CALL.
C
C PRIOR TO CALLING GAUSS4, THE RANDOM NUMBERS IN ARRAYS TR AND TI MUST
C BE INITIALIZED WITH A CALL TO GAUSSI.
C
      DIMENSION A(1),XR(1),XI(1)
      COMMON /GS/ IRSET,NTR,NTI,NMAX,KR,KI,TR(1253),TI(1353)
      IF(N.GT.NMAX) STOP
      DO 35 I=1,N
      XR(I)=A(I)*TR(KR)
      XI(I)=A(I)*TI(KI)
      KR=KR+1
      KI=KI+1
   35 CONTINUE
      IF(IRSET.NE.1) GO TO 40
      KR=KR-N
      KI=KI-N
   40 IF(KR.GT.NTR) KR=KR-NTR
      IF(KI.GT.NTI) KI=KI-NTI
      RETURN
      END
```